WHAT IS CLAIMED IS:

1. A method for compiling an application program in an optimum manner, comprising:

compiling said application with a different set of compiler options to provide two or more executables,

generating profile information from said executables, and

selecting compiler options for said application program based upon said profile information.

2. The method of claim 1, wherein said set of computer options is selected from speed, code size, and power.

3. A method for compiling an application program in an optimum manner, comprising:

compiling said application program with a first set of compiler options to provide a first executable,

compiling said application program with a second set of compiler options to provide a second executable, generating profile information from said first and second executables, and

selecting compiler options for each function of said application program so as to optimize said application program as a function of desired profile information.

4. The method of claim 3, wherein said first set of options is for speed.

5. The method of claim 3, wherein said second set of options is for code size.

6. The method of claim 3, further comprising, analyzing said profile information against user supplied constraints for selecting said compiler options by function.

7. A solution space generator, comprising:

means for reading profiler information, and

means for generating useful solutions of a solution set derived from profiler information.

8. The generation of claim 7, further comprising:

means for providing a display of said useful solutions.

9. The generation of claim 8, further comprising:

means for selecting one of said solutions and using said solution for subsequent compile of an application program.

10. A method for compiling an application program comprising the steps of: computing said application with a different set of compiler options to provide two or more executables; generating profile information from said executables; applying said profile information to a solver; generating sets of useful solutions from said profile information wherein the sets have methods of compiling at the function level; and selecting compiler options for said application program using said useful solutions for subsequent compiling of said application.

11. The method of Claim 10 wherein said selecting step includes displaying said useful solutions.

12. The method of Claim 10 wherein said generating step includes generating an efficient frontier curve of optimum solution points and displaying said curve of solution points.

13. The method of Claim 12 wherein said generating step includes a zoom window of a section of said curve of solution points.

14. The method of Claim 10 wherein said generating step includes linear programming and heuristics to reduce the number of permutations of option sets per function.

15. The method of Claim 10 wherein said generating solutions step generates possible solutions step generates possible solutions and filters the possible solutions.

16. The method of Claim 10 wherein said generating solutions includes a search tree wherein each candidate is applied to a node and compared to the solution at the node and if faster in time and smaller in size replacing that candidate at the node, if neither faster nor smaller in size discarding the candidate, if faster only processing down the tree in one direction and if smaller only processing down the tree in a different direction.

17. The method of Claim 10 wherein the step of selecting compiling option includes displaying a solution point on said solution point curve illustrating for each function the compiling information of size and cycles and method of compiling.

18. The method of Claim 10 including means for displaying a solution point on said solution point curve and means for displaying and overriding a compiling function solution after displaying a selected solution point and thereafter redisplaying the results.

19. The method of Claim 17 including the step after compiling of displaying by function the difference between the expected results and the actual results.

20. A user interface for displaying and controlling the results of compiling an application program with a compiler having a preselected number of compiling options, comprising:

a module for displaying at least a portion of solution information as a function of said selected compiling options and for selecting at least one displayed solution, and

a module for outputting, for a selected solution, compiler information to allow for said application program to be compiled in a manner consistent with said selected solution.

21. A user interface for displaying and controlling the results of compiling an application program with a compiler having a preselected number of compiling options, comprising:

a module for displaying at least a portion of information as a function of selected performance metrics and for selecting at least one displayed solution, and

a module for outputting, for a selected solution, compiler information to allow for said application program to be compiled in a manner consistent with said selected solution.

22. The user interface of claim 21, further comprising:

a module for selecting and fixing instructions for a solver that generates said solutions information.

23. The user interface of claim 22, further comprising:

a module for compiling said application program in a manner consistent with said selected solution and for comparing the results with said selected solution.